Newton-Cotes cubature rules over $(d+1)$-pencil lattices

Jernej Kozak [a], Vito Vitrih [b,*]

[a] *University of Ljubljana, FMF and IMFM, Jadranska 19, Ljubljana, Slovenia*

[b] *University of Primorska, PINT, Muzejski trg 2, Koper, Slovenia*

Abstract

In this paper, Newton-Cotes cubature rules are extended to $(d+1)$-pencil lattices over simplices and simplicial partitions. The closed form of the cubature rules as well as the error term are determined. Further, the basic cubature rules can be combined with an adaptive algorithm over simplicial partitions. The key point of the algorithm is a subdivision step that refines a $(d+1)$-pencil lattice over a simplex to its subsimplices. If the number of function evaluations is crucial, the additional freedom provided by $(d+1)$-pencil lattices may be used to decrease it significantly.

Key words: Integration, Multivariate, Simplex, Lattice, Adaptive

1 Introduction

The multivariate integration has been quite a challenge in numerical analysis since integrals, encountered in many mathematical models, can rarely be calculated analytically. Multivariate integration comes across in practical applications, such as finite elements methods, statistical models, computer graphics, financial mathematics, etc. A cubature rule over a simplex $\triangle \subset \mathbb{R}^d$ of the form

$$Q_{\triangle}(f) = \sum_{\boldsymbol{\gamma}} \omega_{\boldsymbol{\gamma}} f(X_{\boldsymbol{\gamma}}), \quad X_{\boldsymbol{\gamma}} \in \triangle, \tag{1}$$

where $f(X_{\boldsymbol{\gamma}})$ are the values of the function f at points $X_{\boldsymbol{\gamma}}$, $\omega_{\boldsymbol{\gamma}}$ are the weights, and $\boldsymbol{\gamma}$ is a multiindex, is one of the usual ways how to approximate a multivariate integral over a compact domain in $\mathbb{R}^d$. The choice of points $X_{\boldsymbol{\gamma}}$ and

* Corresponding author.
Email address: vito.vitrih@upr.si (Vito Vitrih).

weights $\omega_{\boldsymbol{\gamma}}$ usually does not depend on the function f. There are several criteria to classify cubature rules based on their behavior for specific classes of functions (see [3], e.g.). Probably the most often used rules of the form (1) are polynomial-based ones, which are exact for a particular set of polynomials. In this case, the points $X_{\boldsymbol{\gamma}}$ should provide a basis for the correct interpolation with the polynomial class concerned. If integration points are to be determined in advance, as is the case with Newton-Cotes cubature rules, this is not a trivial job in the multivariate case. A quite well-known approach to obtain points that admit correct interpolation is to use lattices. The principal lattices, where the points are obtained as intersections of $d+1$ pencils of parallel hyperplanes ([10], [2]), lead to the Newton-Cotes cubature rules that can be viewed as a straightforward generalization of the equidistant univariate case. These Newton-Cotes rules can already be found in [12]. Principal lattices have been further generalized to intersections of more general hyperplanes. These lattices are known as $(d+1)$-*pencil lattices of order* n and were introduced in [9]. Even though generalized lattices are nowadays a quite important item in multivariate polynomial interpolation, since they admit correct interpolation problems (see [1], e.g.), their impact on Newton-Cotes numerical integration is not well understood. This is perhaps due to the fact that it was not clear until [7] how to continuously extend a lattice from a particular simplex to its neighbours.

In this paper, Newton-Cotes cubature rules over principal lattices are carried over to $(d+1)$-pencil lattices. The generalization is based upon a simple form of the Lagrange basis polynomials in the barycentric representation. A similar form of the Newton basis polynomials enables us to derive a closed form of the error remainder too. Moreover, by some recent results ([5], [6], [7], [13]), a $(d+1)$-pencil lattice can be extended from a simplex to a simplicial partition, such that the local lattices agree on common faces of the underlying simplices. Thus it is possible to efficiently extend the rules to $(d+1)$-pencil lattices on simplicial partitions. Since usually most of the lattice points lie on facets of simplices, it is therefore very important to evaluate the function f at these points only once. As a bonus, if the function as a mapping is known too, we can improve the approximation by using an adaptive algorithm. Therefore a subdivision step that refines a $(d+1)$-pencil lattice over a simplex to its subsimplices is presented. Moreover, if the number of function evaluations is at stake, the additional freedom of $(d+1)$-pencil lattices can be exploited to obtain a more efficient adaptive algorithm over simplicial partitions.

The extended Newton-Cotes cubature rules are useful in many practical applications. Suppose that the function values over a $(d+1)$-pencil lattice on a simplicial partition are known in advance (for example, they were computed for the construction of a continuous interpolant over the lattice). Then these values should be used also for the numerical integration over the simplicial partition. We can further apply an adaptive algorithm based on the extended

Newton-Cotes rules in order to improve the obtained approximation. Moreover, the cubature rules over $(d+1)$-pencil lattices can be used if the evaluation of a function is much more expensive over some particular parts of a simplicial partition. The additional freedom of the generalized lattices can be used to diminish the number of points on the undesired parts.

The paper is organized as follows. In the next section, some properties of a $(d+1)$-pencil lattice are briefly recalled. In Section 3, Newton-Cotes cubature rules based on $(d+1)$-pencil lattices are derived. In Section 4, a refinement of a $(d+1)$-pencil lattice is introduced as a necessary tool for the adaptive integration, and in the last section, an adaptive cubature rule is presented, together with some numerical examples.

2 (d+1)-pencil lattices

In this section, $(d+1)$-pencil lattices as well as their barycentric representation, introduced in [6], are briefly recalled. A $(d+1)$-pencil lattice of order n on a simplex $\triangle := \langle \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \rangle$ is a set of $\binom{n+d}{d}$ points, generated by $d+1$ pencils of $n+1$ hyperplanes. Any lattice point is an intersection of $d+1$ hyperplanes, one from each pencil. All hyperplanes of the same pencil intersect at a center $\boldsymbol{C}_i \subset \mathbb{R}^d$, $i = 0, 1, \ldots, d$, a plane of codimension two. The lattice is actually determined by $d+1$ lattice control points $\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_d$, $\boldsymbol{P}_i \in \mathbb{R}^d$, where $\boldsymbol{P}_i$ lies on the line passing through the vertices $\boldsymbol{T}_i$ and $\boldsymbol{T}_{i+1}$, but outside of the segment $\boldsymbol{T}_i\boldsymbol{T}_{i+1}$ (Fig. 1). The center $\boldsymbol{C}_i$ is then the unique plane of codimension two that passes through all the control points $\boldsymbol{P}_i, \boldsymbol{P}_{i+1}, \ldots, \boldsymbol{P}_{i+d-2}$. If $d = 2$, the centers $\boldsymbol{C}_i$ are simply the control points $\boldsymbol{P}_i$ (Fig. 1, left). If $d > 2$, more control points are needed (Fig. 1, right), and some of the indices involved are outside of $\{0, 1, \ldots, d\}$. Obviously, in the sequence $\boldsymbol{P}_d$ is followed by $\boldsymbol{P}_0$, etc. Thus, in order to make the discussion as short as possible, indices of control points, vertices, centers, lattice parameters and variables are assumed to be taken modulo $d+1$ throughout the paper.

A closed form of a lattice point has to depend on positions of the control points. In the barycentric form, coordinates of $\boldsymbol{P}_i$ w.r.t. $\boldsymbol{T}_i\boldsymbol{T}_{i+1}$ are particularly simple. Let us denote them

$$\left(\frac{1}{1-\xi_i}, -\frac{\xi_i}{1-\xi_i} \right).$$

Quite clearly, $\xi_i > 0$, since $\boldsymbol{P}_i$ is not lying on the line segment $\boldsymbol{T}_i\boldsymbol{T}_{i+1}$. The range $0 < \xi_i < 1$ covers positions from the ideal line to the vertex $\boldsymbol{T}_i$, and $1 < \xi_i < \infty$ the half-line from $\boldsymbol{T}_{i+1}$ to the ideal line. This reveals the barycentric coordinates of the control points w.r.t. $\triangle$ ([6]) as

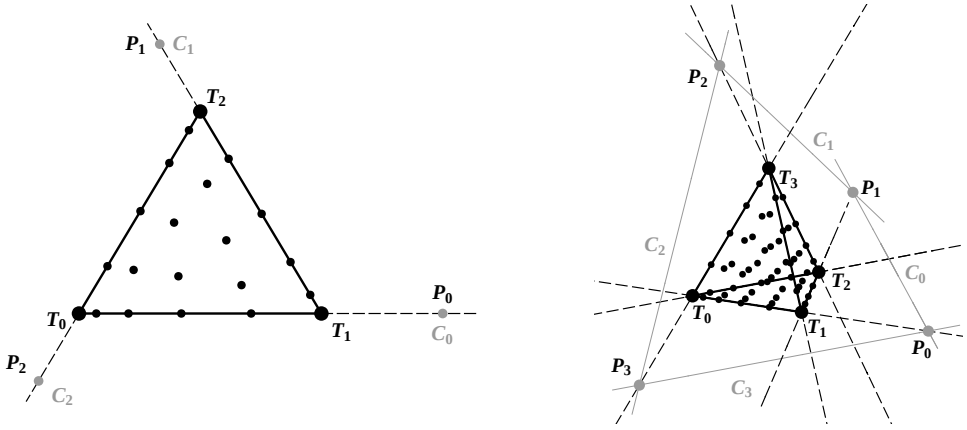

Fig. 1. $(d{+}1)$-pencil lattices with their control points $\boldsymbol{P}_i$ and centers $\boldsymbol{C}_i$ on simplices $\langle \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \rangle$, for $d = 2, 3$.

$$\boldsymbol{P}_i = \left(\underbrace{0, 0, \ldots, 0}_{i}, \frac{1}{1-\xi_i}, -\frac{\xi_i}{1-\xi_i}, \underbrace{0, 0, \ldots, 0}_{d-1-i} \right), \quad i = 0, 1, \ldots, d-1,$$

$$\boldsymbol{P}_d = \left(-\frac{\xi_d}{1-\xi_d}, \underbrace{0, 0, \ldots, 0}_{d-1}, \frac{1}{1-\xi_d} \right).$$

If all of the control points that determine the center $\boldsymbol{C}_i$ are on the ideal line, so is $\boldsymbol{C}_i$, and the corresponding hyperplanes are parallel. The barycentric coordinates of a $(d+1)$-pencil lattice on $\triangle$ w.r.t. $\triangle$ are then determined by $d+1$ parameters $\boldsymbol{\xi}(\triangle) := \boldsymbol{\xi} := (\xi_0, \xi_1, \ldots, \xi_d)$ as

$$B_{\boldsymbol{\gamma}} := B_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) = \frac{1}{\chi_{\boldsymbol{\gamma},\boldsymbol{\xi}}} \left(\alpha^{n-\gamma_0} [\gamma_0]_\alpha, \xi_0 \alpha^{n-\gamma_0-\gamma_1} [\gamma_1]_\alpha, \ldots, \xi_0 \xi_1 \cdots \xi_{d-1} [\gamma_d]_\alpha \right), \tag{2}$$

with

$$\chi_{\boldsymbol{\gamma},\boldsymbol{\xi}} := \alpha^{n-\gamma_0} [\gamma_0]_\alpha + \xi_0 \alpha^{n-\gamma_0-\gamma_1} [\gamma_1]_\alpha + \ldots + \xi_0 \xi_1 \cdots \xi_{d-1} [\gamma_d]_\alpha,$$

where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_d) \in \mathbb{N}_0^{d+1}$, $|\boldsymbol{\gamma}| := \sum_{i=0}^{d} \gamma_i = n$,

$$\alpha := \sqrt[n]{\prod_{i=0}^{d} \xi_i} > 0, \quad \text{and} \quad [j]_\alpha := \begin{cases} \dfrac{1-\alpha^j}{1-\alpha}, & \alpha \neq 1, \\ j, & \alpha = 1, \end{cases} \quad j \in \mathbb{N}_0.$$

The indices $\boldsymbol{\gamma}$ in (2) are determined by hyperplanes $H_{i,j}$ such that

$$B_{\boldsymbol{\gamma}} := \bigcap_{i=0}^{d} H_{i,\gamma_i}.$$

Here $H_{i,j}$ is the $(j+1)$-th hyperplane passing through the center $\boldsymbol{C}_{i+1}$. Since $|\boldsymbol{\gamma}| = n$, one can drop any fixed component of the index, and the lattice

points will still be uniquely denoted. So, with $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_d)$, and $\boldsymbol{\gamma}' = (\gamma_1, \gamma_2, \ldots, \gamma_d)$,

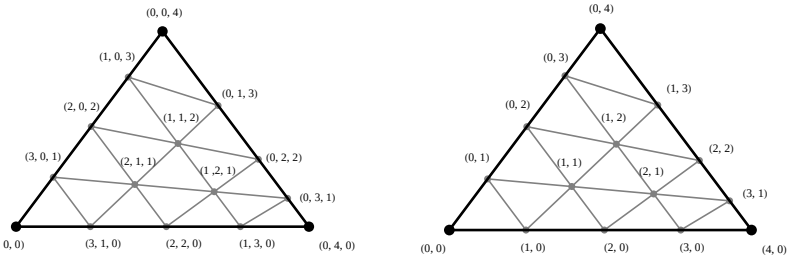

Fig. 2. The indices of lattice points: $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \ldots, \gamma_d)$, $|\boldsymbol{\gamma}| = n$, (left), and $\boldsymbol{\gamma}' = (\gamma_1, \gamma_2, \ldots, \gamma_d)$, $|\boldsymbol{\gamma}'| \leq n$, (right).

$$\{B_{\boldsymbol{\gamma}},\ \boldsymbol{\gamma} \in \mathbb{N}_0^{d+1}, |\boldsymbol{\gamma}| = n\} \quad \text{and} \quad \{B_{\boldsymbol{\gamma}'},\ \boldsymbol{\gamma}' \in \mathbb{N}_0^{d}, |\boldsymbol{\gamma}'| \leq n\} \tag{3}$$

refer to the same set of points (Fig. 2).

3 Newton-Cotes cubature rules for a simplex

Throughout the paper, $S_{\triangle}(f)$ will denote the integral of a scalar field $f : \triangle \to \mathbb{R}$ over a simplex $\triangle$. The cubature rules will be based on the barycentric form. As expected, this will enable us to extend the rules to an arbitrary simplex in $\mathbb{R}^d$ by a simple transformation only. If $\boldsymbol{\gamma} \in \mathbb{N}_0^{d+1}$, then let

$$Q(f) := Q^{(n)}(f; \boldsymbol{\xi}) := \sum_{|\boldsymbol{\gamma}|=n} \omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) f_{\boldsymbol{\gamma}}, \tag{4}$$

where $f_{\boldsymbol{\gamma}}$ is the value of the function f at the point with the barycentric coordinates $B_{\boldsymbol{\gamma}}$, denote a cubature rule of degree n in the barycentric form for the standard simplex $\triangle_d \subseteq \mathbb{R}^{d+1}$,

$$\triangle_d = \langle\, \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \,\rangle, \quad \boldsymbol{T}_i = (\delta_{i,j})_{j=0}^{d},\ i = 0, 1, \ldots, d.$$

Here $\delta_{i,j}$ is the Kronecker delta. But Newton-Cotes rules are interpolatory, so the weights $\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ are determined as

$$\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) = S_{\triangle_d}(\mathcal{L}_{\boldsymbol{\gamma}}), \quad |\boldsymbol{\gamma}| = n, \tag{5}$$

where $\mathcal{L}_{\boldsymbol{\gamma}}$ are the Lagrange basis polynomials in the barycentric form. They have been explicitly determined in [6]. Since $(d+1)$-pencil lattices satisfy the geometric characterization (GC) condition ([2]), these polynomials are of a particularly simple form,

$$\mathcal{L}_{\boldsymbol{\gamma}}(\boldsymbol{x}; \boldsymbol{\xi}) = \prod_{i=0}^{d} \prod_{j=0}^{\gamma_i - 1} \frac{h_{i,j}(\boldsymbol{x}; \boldsymbol{\xi})}{h_{i,j}(B_{\boldsymbol{\gamma}}; \boldsymbol{\xi})}, \quad \boldsymbol{x} := (x_i)_{i=0}^{d} \in \mathbb{R}^{d+1},\ \sum_{i=0}^{d} x_i = 1,$$

with

$$h_{i,j}\left(\boldsymbol{x};\boldsymbol{\xi}\right)=\sum_{t=i}^{i+d}a_{t,j}(\boldsymbol{\xi})\,x_t,\quad a_{t,j}(\boldsymbol{\xi})=\begin{cases}\left[n-j\right]_\alpha, & t=i,\\ \left(\left[n-j\right]_\alpha-\left[n\right]_\alpha\right)\left(\prod_{k=i}^{t-1}\xi_k\right)^{-1}, & t>i.\end{cases}\tag{6}$$

Note that indices of $a_{t,j}(\boldsymbol{\xi})$ are not taken modulo $d+1$. Here $h_{i,j}\left(\boldsymbol{x};\boldsymbol{\xi}\right)=0$ is the equation of the hyperplane $H_{i,j}$ in the barycentric form, based upon the center $\boldsymbol{C}_{i+1}$ and lattice points $B_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$, $\boldsymbol{\gamma}\in\mathbb{N}_0^{d+1}$, $|\boldsymbol{\gamma}|=n$, $\gamma_i=j$.

Let us introduce the sets of indices $\Lambda^i_{\boldsymbol{\gamma}}$, $i=0,1,\ldots,d$, and $\Lambda_{\boldsymbol{\gamma}}$ (Fig. 3). If $\gamma_i\neq 0$, then

$$\Lambda^i_{\boldsymbol{\gamma}}:=\left\{\boldsymbol{\lambda}^i:=\left(\lambda^i_0,\ldots,\lambda^i_{\gamma_i-1}\right),\ \lambda^i_0=i,\ \lambda^i_j\in\{i,\ldots,i+d\},\ 0<j\leq\gamma_i-1\right\},$$

otherwise $\Lambda^i_{\boldsymbol{\gamma}}:=\emptyset$. Further,

$$\Lambda_{\boldsymbol{\gamma}}:=\left\{\boldsymbol{\lambda}:=\left(\lambda^0_0,\ldots,\lambda^0_{\gamma_0-1},\ldots,\lambda^d_0,\ldots,\lambda^d_{\gamma_d-1}\right)\in\mathbb{N}_0^n,\quad\left(\lambda^i_0,\ldots,\lambda^i_{\gamma_i-1}\right)\in\Lambda^i_{\boldsymbol{\gamma}}\right\}.$$

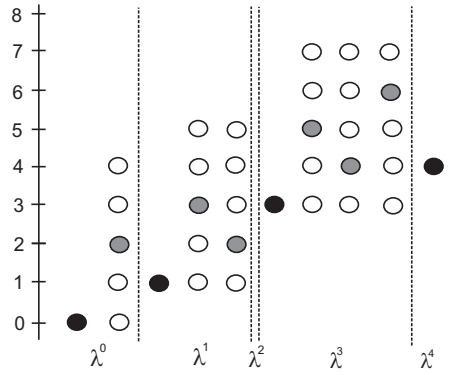

Fig. 3. $\boldsymbol{\lambda}=(0,2,1,3,2,3,5,4,6,4)\in\Lambda_{\boldsymbol{\gamma}}$ is an example for $d=4$, $n=10$ and $\boldsymbol{\gamma}=(2,3,0,4,1)$. Any other selection of grey points would determine another index vector in $\Lambda_{\boldsymbol{\gamma}}$.

We can now state the following theorem.

Theorem 1 *The weights $\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ of the cubature rule* (4) *are*

$$\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi})=K(\boldsymbol{\xi})\cdot\sum_{\boldsymbol{\lambda}\in\Lambda_{\boldsymbol{\gamma}}}\left(\prod_{i=0}^{d}\prod_{j=0}^{\gamma_i-1}a_{\lambda^i_j,j}(\boldsymbol{\xi})\right)\frac{k_{\boldsymbol{\lambda},0}!\,k_{\boldsymbol{\lambda},1}!\cdots k_{\boldsymbol{\lambda},d}!}{(n+d)!},\tag{7}$$

where

$$K(\boldsymbol{\xi}):=\left(\prod_{i=0}^{d}\prod_{j=0}^{\gamma_i-1}h_{i,j}(B_{\boldsymbol{\gamma}};\boldsymbol{\xi})\right)^{-1},$$

and $k_{\boldsymbol{\lambda},i}$ denotes the frequency of i in $\boldsymbol{\lambda}$.

PROOF. By (5),

$$\begin{aligned}\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) = S_{\triangle_d}(\mathcal{L}_{\boldsymbol{\gamma}}) \;&=\; K(\boldsymbol{\xi}) \cdot \int_{\triangle_d} \left(\prod_{i=0}^{d} \prod_{j=0}^{\gamma_i - 1} h_{i,j}(\boldsymbol{x}; \boldsymbol{\xi}) \right) \mathrm{d}\boldsymbol{x} = \\ &= K(\boldsymbol{\xi}) \cdot \int_{\triangle_d} \prod_{i=0}^{d} \left(\prod_{j=0}^{\gamma_i - 1} \sum_{t=i}^{i+d} a_{t,j}(\boldsymbol{\xi}) x_t \right) \mathrm{d}\boldsymbol{x} = \\ &= K(\boldsymbol{\xi}) \cdot \int_{\triangle_d} \prod_{i=0}^{d} \left(\sum_{\boldsymbol{\lambda}^i \in \Lambda^i_{\boldsymbol{\gamma}}} \prod_{j=0}^{\gamma_i - 1} a_{\lambda^i_j, j}(\boldsymbol{\xi})\, x_{\lambda^i_j} \right) \mathrm{d}\boldsymbol{x} = \\ &= K(\boldsymbol{\xi}) \cdot \sum_{\boldsymbol{\lambda} \in \Lambda_{\boldsymbol{\gamma}}} \left(\prod_{i=0}^{d} \prod_{j=0}^{\gamma_i - 1} a_{\lambda^i_j, j}(\boldsymbol{\xi}) \cdot \int_{\triangle_d} \prod_{i=0}^{d} \prod_{j=0}^{\gamma_i - 1} x_{\lambda^i_j} \, \mathrm{d}\boldsymbol{x} \right).\end{aligned}$$

With the help of the notation $k_{\boldsymbol{\lambda},i}$ we count the multiplicity of x_i in the product $\prod_{i=0}^{d} \prod_{j=0}^{\gamma_i - 1} x_{\lambda^i_j}$ and obtain

$$\prod_{i=0}^{d} \prod_{j=0}^{\gamma_i - 1} x_{\lambda^i_j} = \prod_{i=0}^{d} \left(\prod_{\ell=0}^{d} x_\ell^{k_{\boldsymbol{\lambda}^i, \ell}} \right) = \prod_{\ell=0}^{d} x_\ell^{k_{\boldsymbol{\lambda}, \ell}}.$$

Now, $\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ becomes

$$\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) = K(\boldsymbol{\xi}) \cdot \sum_{\boldsymbol{\lambda} \in \Lambda_{\boldsymbol{\gamma}}} \left(\prod_{i=0}^{d} \prod_{j=0}^{\gamma_i - 1} a_{\lambda^i_j, j}(\boldsymbol{\xi}) \int_{\triangle_d} \prod_{i=0}^{d} x_i^{k_{\boldsymbol{\lambda}, i}} \, \mathrm{d}\boldsymbol{x} \right). \tag{8}$$

Further, with Γ being the gamma function,

$$\int_{\triangle_d} \prod_{i=0}^{d} x_i^{k_{\boldsymbol{\lambda}, i}} \, \mathrm{d}\boldsymbol{x} = \frac{\Gamma(k_{\boldsymbol{\lambda},0} + 1)\, \Gamma(k_{\boldsymbol{\lambda},1} + 1) \cdots \Gamma(k_{\boldsymbol{\lambda},d} + 1)}{\Gamma(k_{\boldsymbol{\lambda},0} + k_{\boldsymbol{\lambda},1} + \ldots + k_{\boldsymbol{\lambda},d} + d + 1)},$$

where for $\boldsymbol{x} \in \mathbb{R}^{d+1}$, $\sum_{i=0}^{d} x_i = 1$,

$$\int_{\triangle_d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} := \int_0^1 \mathrm{d}x_1 \int_0^{1-x_1} \mathrm{d}x_2 \cdots \int_0^{1-\sum_{i=1}^{d} x_i} f((1 - \sum_{i=1}^{d} x_i,\, x_1, \ldots, x_d)) \, \mathrm{d}x_d. \tag{9}$$

Since

$$\sum_{i=0}^{d} k_{\boldsymbol{\lambda}, i} = n,$$

it follows

$$\int_{\triangle_d} \prod_{i=0}^{d} x_i^{k_{\boldsymbol{\lambda}, i}} \, \mathrm{d}\boldsymbol{x} = \frac{k_{\boldsymbol{\lambda},0}!\, k_{\boldsymbol{\lambda},1}! \cdots k_{\boldsymbol{\lambda},d}!}{(n+d)!},$$

and the proof is concluded. □

As an example, let us compute the weights for $d = 2$ and $n = 3$. The barycentric form of the cubature rule is given by

$$Q^{(3)}(f;\boldsymbol{\xi}) := \sum_{|\boldsymbol{\gamma}|=3} \omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) f_{\boldsymbol{\gamma}}, \qquad \boldsymbol{\gamma} \in \mathbb{N}_0^3,$$

and $\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$, $|\boldsymbol{\gamma}| = 3$, is equal to one of the following possibilities:

$$\begin{aligned}
&\boldsymbol{\gamma} \in \{(3,0,0),(0,3,0),(0,0,3)\},\ i := (\gamma_i = 3) : \\
&\quad \frac{1}{20} + \frac{\alpha^3(1+\alpha) + \alpha\xi_{i+1}(1+\xi_{i+1})\left(\alpha^2(1+\alpha) - (1+\alpha(3+\alpha))\xi_i\right)}{60\,(1+\alpha)\xi_i^2\xi_{i+1}^2}, \\
&\boldsymbol{\gamma} \in \{(2,1,0),(2,0,1),(0,2,1)\},\ i := (\gamma_i = 2),\ j := (\gamma_j = 1) : \\
&\quad -\frac{\left(\alpha + \alpha^2 + \prod_{t=i}^{j-1}\xi_t\right)^3 \left((j-i)^2\alpha^2 + 2\xi_{i+1}\left(\alpha^2 - (j-i)(1+\alpha)\xi_i\right)\right)}{(j-i)\,120\,\alpha^2(1+\alpha)(1+\alpha+\alpha^2)\xi_i^2\xi_{i+1}^{j-i}}, \\
&\boldsymbol{\gamma} \in \{(1,2,0),(1,0,2),(0,1,2)\},\ i := (\gamma_i = 2),\ j := (\gamma_j = 1) : \\
&\quad -\frac{\left(\alpha^2 + (1+\alpha)\prod_{t=j}^{i-1}\xi_t\right)^3 \left(2\alpha^2 + (i-j)\xi_{i+1}\left((i-j)\alpha^2 - 2(1+\alpha)\xi_i\right)\right)}{(i-j)\,120\,\alpha^5(1+\alpha)\left(\alpha - (1+\alpha)^2\right)\xi_j^{i-j}\xi_1^{i-j-1}}, \\
&\boldsymbol{\gamma} = (1,1,1) : \\
&\quad \frac{\left(\alpha^2 + \xi_0(\alpha+\xi_1)\right)^3}{120\,\alpha^3\xi_0^2\xi_1}.
\end{aligned}$$

Consider now a simplex $\triangle = \langle \boldsymbol{V}_0, \boldsymbol{V}_1, \ldots, \boldsymbol{V}_d \rangle \subset \mathbb{R}^d$. Let $X_{\boldsymbol{\gamma}}$, $\boldsymbol{\gamma} \in \mathbb{N}_0^{d+1}$, $|\boldsymbol{\gamma}| = n$, denote the Cartesian coordinates of lattice points. They are obtained from the barycentric representation as

$$X_{\boldsymbol{\gamma}} = \sum_{j=0}^{d} (B_{\boldsymbol{\gamma}})_{j+1}\ \boldsymbol{V}_j.$$

We are now able to state the following corollary.

Corollary 2 *A Newton-Cotes cubature rule of degree n for a simplex $\triangle \subset \mathbb{R}^d$ is*

$$Q_{\triangle}(f) := Q_{\triangle}^{(n)}(f;\boldsymbol{\xi}) := \sum_{|\boldsymbol{\gamma}|=n} \omega_{\boldsymbol{\gamma},\triangle}(\boldsymbol{\xi})\, f(X_{\boldsymbol{\gamma}}) = d!\,\mathrm{vol}\,(\triangle) \sum_{|\boldsymbol{\gamma}|=n} \omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi})\, f(X_{\boldsymbol{\gamma}}),$$

where $\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ are the weights given by (7) *and* $\mathrm{vol}\,(\triangle)$ *is the volume of the simplex $\triangle$.*

PROOF. Let $\widetilde{\triangle} := \langle \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \rangle \subset \mathbb{R}^d$, $\boldsymbol{T}_i = (\delta_{i,j})_{j=1}^d$, and let $\tilde{\boldsymbol{u}} :=$

$(\tilde{u}_i)_{i=1}^d \in \widetilde{\triangle}$. Then the barycentric coordinates of $\tilde{\boldsymbol{u}}$ w.r.t. $\widetilde{\triangle}$ are

$$\left(1-\sum_{i=1}^{d}\tilde{u}_i,\tilde{u}_1,\ldots,\tilde{u}_d\right) =: (\tilde{u}_0,\tilde{\boldsymbol{u}}).$$

Using (8) and (9), we obtain

$$\omega_{\boldsymbol{\gamma},\widetilde{\triangle}}(\boldsymbol{\xi}) = \omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}).$$

Suppose now $\triangle \subset \mathbb{R}^d$ is an arbitrary simplex, and let $\boldsymbol{u} := (u_i)_{i=1}^d \in \triangle$. Further, let $\boldsymbol{x}(\boldsymbol{u}) := (x_i(\boldsymbol{u}))_{i=0}^d$ be the barycentric coordinates of $\boldsymbol{u}$ w.r.t. $\triangle$. By the definition of barycentric coordinates, $(x_i(\boldsymbol{u}))_{i=0}^d = (\tilde{u}_i)_{i=0}^d$. Since

$$\omega_{\boldsymbol{\gamma},\triangle}(\boldsymbol{\xi}) = S_{\triangle}(\mathcal{L}_{\boldsymbol{\gamma}}) = K(\boldsymbol{\xi})\cdot\sum_{\boldsymbol{\lambda}\in\Lambda_{\boldsymbol{\gamma}}}\left(\prod_{i=0}^{d}\prod_{j=0}^{\gamma_i-1}a_{\lambda_j^i,j}(\boldsymbol{\xi})\int_{\triangle}\prod_{i=0}^{d}x_i(\boldsymbol{u})^{k_{\boldsymbol{\lambda},i}}\,\mathrm{d}\boldsymbol{u}\right)$$

and

$$\int_{\widetilde{\triangle}}\prod_{i=0}^{d}\tilde{u}_i^{k_{\boldsymbol{\lambda},i}}\,\mathrm{d}\tilde{\boldsymbol{u}} = J\cdot\int_{\triangle}\prod_{i=0}^{d}x_i(\boldsymbol{u})^{k_{\boldsymbol{\lambda},i}}\,\mathrm{d}\boldsymbol{u},\ J=\det\left(\frac{\partial\tilde{\boldsymbol{u}}}{\partial\boldsymbol{u}}\right)=\frac{\mathrm{vol}(\widetilde{\triangle})}{\mathrm{vol}(\triangle)}=\frac{1}{d!\,\mathrm{vol}(\triangle)},$$

where $\frac{\partial\tilde{\boldsymbol{u}}}{\partial\boldsymbol{u}}$ is the Jacobian matrix, it follows

$$\omega_{\boldsymbol{\gamma},\triangle}(\boldsymbol{\xi}) = \frac{1}{J}\,\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) = d!\,\mathrm{vol}(\triangle)\,\omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}).$$

The proof is concluded. □

Our next goal is to derive the error term of the cubature rule (4) in the barycentric form for a sufficiently smooth function f. Let us recall that the Lagrange interpolating polynomial, which interpolates given data $f_{\boldsymbol{\gamma}} \in \mathbb{R}$, $\boldsymbol{\gamma} \in \mathbb{N}_0^{d+1}$, $|\boldsymbol{\gamma}| = n$, at the $(d+1)$-pencil lattice points with barycentric coordinates $(B_{\boldsymbol{\gamma}})_{|\boldsymbol{\gamma}|=n}$, is

$$p_n(\boldsymbol{x};\boldsymbol{\xi}) = \sum_{|\boldsymbol{\gamma}|=n} f_{\boldsymbol{\gamma}}\,\mathcal{L}_{\boldsymbol{\gamma}}(\boldsymbol{x};\boldsymbol{\xi}),\quad \boldsymbol{x}\in\mathbb{R}^{d+1},\ \sum_{i=0}^{d}x_i = 1.$$

The error of a cubature rule is then obtained as $S_{\triangle_d}(f-p_n)$. So we have to derive the interpolation error in a convenient form first. We shall follow the way paved in [11] and [1]. But then the Newton basis polynomials in the barycentric form need to be determined, too. These are the polynomials of total degrees $|\boldsymbol{\gamma}| \le n$, that vanish at particular subsets of interpolation points. In order to determine these sets precisely, let us use the abbreviated lattice point indexation introduced in (3) (see Fig. 2). Since the Newton polynomials

$\mathcal{N}_{\boldsymbol{\gamma}'}$ satisfy ([1])

$$\mathcal{N}_{\boldsymbol{\gamma}'}(B_{\boldsymbol{\beta}'}) = \delta_{\boldsymbol{\gamma}',\boldsymbol{\beta}'}, \quad \forall\, \boldsymbol{\gamma}', \boldsymbol{\beta}' \in \mathbb{N}_0^d,\ |\boldsymbol{\beta}'| \le |\boldsymbol{\gamma}'| \le n,$$

they also have a very simple barycentric representation

$$\mathcal{N}_{\boldsymbol{\gamma}'}(\boldsymbol{x};\boldsymbol{\xi}) = \prod_{i=1}^{d} \prod_{j=0}^{\gamma_i - 1} \frac{h_{i,j}(\boldsymbol{x};\boldsymbol{\xi})}{h_{i,j}(B_{\boldsymbol{\gamma}'};\boldsymbol{\xi})}.$$

This follows from the facts that a hyperplane $H_{i,j}$ with the equation $h_{i,j} = 0$, given by (6), vanishes at lattice points $B_{\boldsymbol{\beta}'}$, $\beta_i = j$, and that for $\boldsymbol{\beta}' \neq \boldsymbol{\gamma}', |\boldsymbol{\beta}'| \le |\boldsymbol{\gamma}'|$, there exists an index $i \in \{1, 2, \ldots, d\}$, such that $\beta_i < \gamma_i$. Further, let us recall some notation from [1] and translate it to the barycentric form. Let Ξ_n represent the set of all paths

$$\underline{\boldsymbol{\mu}} = (\boldsymbol{\mu}'_0, \boldsymbol{\mu}'_1, \ldots, \boldsymbol{\mu}'_n), \quad \boldsymbol{\mu}'_j \in \mathbb{N}_0^d,\ |\boldsymbol{\mu}'_j| = j,\ j = 0, 1, \ldots, n.$$

With any path $\underline{\boldsymbol{\mu}}$, let us associate a set of lattice points $B_{\underline{\boldsymbol{\mu}}}$, a corresponding n-th order differential operator $D^n_{\underline{\boldsymbol{\mu}}}$, and a number $\Pi_{\underline{\boldsymbol{\mu}}}$,

$$B_{\underline{\boldsymbol{\mu}}} := B_{\underline{\boldsymbol{\mu}}}(\boldsymbol{\xi}) := \{B_{\boldsymbol{\mu}'_0}, B_{\boldsymbol{\mu}'_1}, \ldots, B_{\boldsymbol{\mu}'_n}\},$$

$$D^n_{\underline{\boldsymbol{\mu}}} := D^n_{\underline{\boldsymbol{\mu}}}(\boldsymbol{\xi}) := D_{B_{\boldsymbol{\mu}'_n} - B_{\boldsymbol{\mu}'_{n-1}}} \cdot D_{B_{\boldsymbol{\mu}'_{n-1}} - B_{\boldsymbol{\mu}'_{n-2}}} \cdots D_{B_{\boldsymbol{\mu}'_1} - B_{\boldsymbol{\mu}'_0}},$$

$$\Pi_{\underline{\boldsymbol{\mu}}} := \Pi_{\underline{\boldsymbol{\mu}}}(\boldsymbol{\xi}) := \prod_{j=0}^{n-1} \mathcal{N}_{\boldsymbol{\mu}'_j}(B_{\boldsymbol{\mu}'_{j+1}};\boldsymbol{\xi}).$$

But the construction of the Newton basis polynomials on $(d+1)$-pencil lattices gives $\Pi_{\underline{\boldsymbol{\mu}}} = 0$ if $\underline{\boldsymbol{\mu}} \notin \widetilde{\Xi}_n$, where

$$\widetilde{\Xi}_n := \{\underline{\boldsymbol{\mu}} \in \Xi_n,\ \boldsymbol{\mu}'_{j+1} = \boldsymbol{\mu}'_j + (\delta_{i,k})_{i=1}^d,\ k \in \{1, 2, \ldots, d\},\ j = 0, 1, \ldots, n-1\}.$$

This reveals the barycentric form of the interpolation error, derived in [11], as

$$f(\boldsymbol{x}) - p_n(\boldsymbol{x};\boldsymbol{\xi}) = \sum_{\underline{\boldsymbol{\mu}} \in \widetilde{\Xi}_n} \mathcal{N}_{\boldsymbol{\mu}'_n}(\boldsymbol{x};\boldsymbol{\xi})\, \Pi_{\underline{\boldsymbol{\mu}}} \int_{[B_{\underline{\boldsymbol{\mu}}},\, \boldsymbol{x}]} D_{\boldsymbol{x} - B_{\boldsymbol{\mu}'_n}} D^n_{\underline{\boldsymbol{\mu}}} f,\ \boldsymbol{x} \in \mathbb{R}^{d+1}, \sum_{i=0}^{d} x_i = 1,$$

where

$$f(\boldsymbol{x}) := f(\boldsymbol{u}(\boldsymbol{x})), \quad \boldsymbol{u}(\boldsymbol{x}) = \sum_{j=0}^{d} x_j\, \boldsymbol{V}_j,$$

and

$$\int_{[\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_m]} f := \int_{\triangle_m} f(t_0 \boldsymbol{z}_0 + t_1 \boldsymbol{z}_1 + \cdots + t_m \boldsymbol{z}_m)\, \mathrm{d}\boldsymbol{t}.$$

This proves the following theorem

Theorem 3 *Let $f \in C^{n+1}(\mathbb{R}^d)$. The barycentric form of the error of the cubature rule* (4) *is given as*

$$E(f) = S_{\triangle_d} \left(\sum_{\underline{\boldsymbol{\mu}} \in \widetilde{\widetilde{\Xi}}_n} \mathcal{N}_{\boldsymbol{\mu}'_n}(\boldsymbol{x}; \boldsymbol{\xi}) \Pi_{\underline{\boldsymbol{\mu}}} \int_{[B_{\underline{\boldsymbol{\mu}}}, \boldsymbol{x}]} D_{\boldsymbol{x} - B_{\boldsymbol{\mu}'_n}} \, D^n_{\underline{\boldsymbol{\mu}}} f \right).$$

4 Lattice refinement

In [7] it was shown that a $(d+1)$-pencil lattice can be extended from a simplex to a simplicial partition. The extension is such that any pair of simplices that share a common face, share the lattice restriction to that face too (see Fig. 4, e.g.). This implies a continuous piecewise polynomial interpolant over the extended lattice. For a regular simplicial partition $\mathcal{T}$ in $\mathbb{R}^d$ with $V \geq d+1$ vertices, there exists a $(d+1)$-pencil lattice on $\mathcal{T}$ which is determined by precisely V parameters. Using this extension, the cubature rule (4) can be efficiently extended from a simplex to a simplicial partition. Since for small enough degrees most of the lattice points lie on facets of simplices, the described extension enables us to evaluate the function at these points only once.

Newton-Cotes cubature rules become really useful in practice when one applies them in an adaptive way. A globally adaptive algorithm over a simplicial partition is usually based upon successive refinements or subdivisions of simplices. Though it is obvious that such a refinement could be carried out for principal lattices, it is far away of being obvious that this can be done for $(d+1)$-pencil lattices too. In this section we present a lattice refinement step that is a basis of the adaptive algorithm in the next section.

The lattice refinement approach is quite useful in multivariate interpolation. Maybe the interpolating polynomial on some simplex of the partition provides too poor approximation. An obvious remedy is to increase the number of interpolation points on this simplex (Fig. 4). A natural way to do this is to refine a lattice. Let $\triangle \in \mathcal{T}$ be the simplex where the lattice refinement is needed. In order to retain regularity of a simplicial partition, let us refine the lattice by adding a new vertex into the interior of $\triangle$. The refinement of a lattice on the simplicial partition $\mathcal{T}$ consists of the following steps (Fig. 4):

- Choose a simplex $\triangle \in \mathcal{T}$, where the refinement is needed.
- Add a new vertex $\boldsymbol{T}$ into the interior of $\triangle$.
- Add $d+1$ edges from $\boldsymbol{T}$ to the vertices of $\triangle$. These edges split the simplex $\triangle$ into $d+1$ new simplices.
- Construct new lattices on these simplices such that two adjacent simplices

share the lattice restriction to the common face.

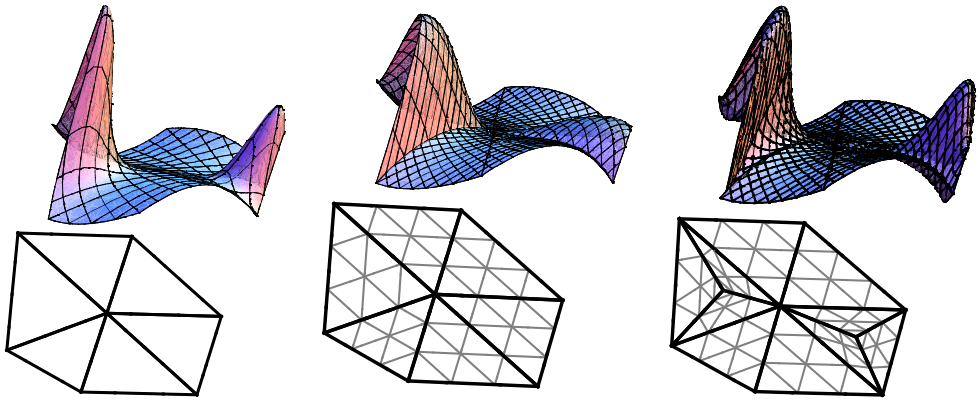

Fig. 4. A given surface and two different continuous piecewise polynomial interpolants over lattices on underlying triangulations.

The following theorem will precisely determine the last step of the lattice refinement. The notation $\mathrm{bd}(\triangle)$ will be used to denote the boundary (the union of all facets) of a simplex $\triangle$.

Theorem 4 *Let $B_{\boldsymbol{\gamma}}(\boldsymbol{\xi})$ be the barycentric coordinates of a $(d+1)$-pencil lattice on $\triangle = \langle \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \rangle$, and let $\boldsymbol{T}_{d+1}$ be a vertex in the interior of $\triangle$ that splits $\triangle$ to $d+1$ simplices $\{\triangle'_i\}_{i=1}^{d+1}$. Then there exist $(d+1)$-pencil lattices on $\{\triangle'_i\}_{i=1}^{d+1}$ which coincide on common faces of $\{\triangle'_i\}_{i=1}^{d+1}$ and agree with the initial lattice on bd($\triangle$). Moreover, there is one degree of freedom to construct these lattices (see Fig. 5).*

PROOF. Let us order the vertices of $\triangle'_i$, $i = 1, 2, \ldots, d+1$, as

$$\triangle'_i = \langle \boldsymbol{T}_{i_0}, \boldsymbol{T}_{i_1}, \ldots, \boldsymbol{T}_{i_d} \rangle, \; 0 \le i_0 < i_1 < \cdots < i_d = d+1. \tag{10}$$

Note that the indices of vertices are not taken modulo $d+1$ here. Any pair of simplices $\triangle'_i$, $\triangle'_j$ has a facet in common. Let this facet be in $\triangle'_i$ denoted as

$$\langle \boldsymbol{T}_{i_{r_0}}, \boldsymbol{T}_{i_{r_1}}, \ldots, \boldsymbol{T}_{i_{r_{d-1}}} \rangle, \quad 0 \le i_{r_0} < i_{r_1} < \cdots < i_{r_{d-1}} \le d+1,$$

with the corresponding vertices in $\triangle'_j$ given by

$$\boldsymbol{T}_{j_{r_k}} = \boldsymbol{T}_{i_{r_k}}, \quad k = 0, 1, \ldots, d-1.$$

In [7] the winding number of an index vector $(v_j)_{j=0}^r$,

$$w\left((v_j)_{j=0}^r\right) := \sum_{k=0}^{r-1} H(v_k - v_{k+1}) + H(v_r - v_0), \quad H(t) := \begin{cases} 1, \; t > 0, \\ 0, \; \text{otherwise}, \end{cases}$$

has been defined. By (10),

$$w\left(\left(i_{r_0}, i_{r_1}, \ldots, i_{r_{d-1}}\right)\right) = w\left(\left(j_{r_0}, j_{r_1}, \ldots, j_{r_{d-1}}\right)\right) = 1. \tag{11}$$

Assume that the product of local barycentric lattice parameters on each simplex in $\{\triangle'_i\}_{i=1}^{d+1}$ is equal to the product of local barycentric lattice parameters for the lattice on $\triangle$. All simplices $\triangle'_i$, $i = 1, 2, \ldots, d+1$, have one facet in common with $\triangle$. Let us first construct the lattice on $\triangle'_1$. Since a similar relation as in (11) holds on the common facet, the lattice can be extended from this facet to $\triangle'_1$ with one additional free parameter ([7], Corollary 7). Now the simplices $\triangle'_i$, $i = 2, 3, \ldots, d+1$, have two facets in common with $\triangle \cup \triangle'_1$. Therefore by the same argument as in the proof of Theorem 8 in [7] all lattices on $\triangle'_2, \triangle'_3, \ldots, \triangle'_{d+1}$ are uniquely determined and agree with the given one on $\mathrm{bd}(\triangle) \cup \mathrm{bd}(\triangle'_1)$. In order to conclude the proof, it only has to be shown that the lattices agree on common facets $\triangle'_{ij} := \triangle'_i \cap \triangle'_j$, $2 \le i < j \le d+1$. Since $\triangle'_{ij}$ are $(d-1)$-simplices, the case $d = 2$ has to be considered separately. For $d \ge 3$, lattices on $\triangle'_{ij}$ are already uniquely determined by the lattices on $\triangle \cup \triangle'_1$ ([7], Corollary 4) and therefore the lattices on $\triangle'_i$ and $\triangle'_j$ agree on $\triangle'_{ij}$. Now let $d = 2$. The same corollary can not be used now, since facets of $(d-1)$-simplices are vertices and they do not include any information about the lattice. However, a direct computation, using Corollary 7 of [7] six times, concludes the proof. □

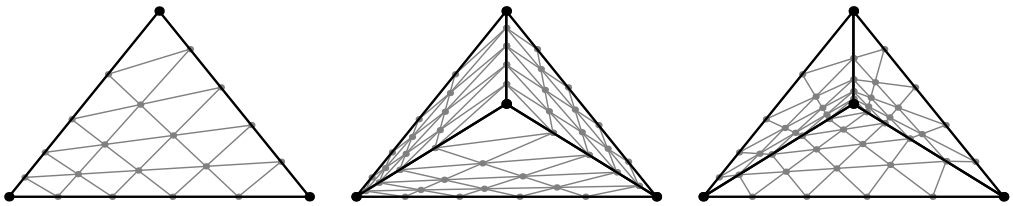

Fig. 5. A lattice with parameters $\xi_0 = 1/2$, $\xi_1 = 3/5$, $\xi_2 = 4/3$, and two different refinements with the additional shape parameter $\zeta = 7/3$, $1/2$, respectively.

5 Adaptive cubature rule

In this section, we study derived cubature rules, applied in an adaptive way. Let us consider the integrals of the form

$$\int_{\mathcal{T}} f(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u} = \sum_{\triangle \in \mathcal{T}} \int_{\triangle} f(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}, \tag{12}$$

where $\mathcal{T}$ is a simplicial partition in $\mathbb{R}^d$, using an adaptive algorithm that consists of a sequence of stages, where each stage has the following steps:

(a) from the current simplicial partition $\mathcal{T}'$ (at the beginning $\mathcal{T}' = \mathcal{T}$) select simplices $\triangle$, where the cubature rule does not give a satisfying approximation,
(b) subdivide selected simplices and determine the lattices on the newly obtained simplices,
(c) update the simplicial partition $\mathcal{T}'$ with new simplices, apply a local cubature rule to any new simplex by carefully avoiding extraneous function evaluations, and update the global integral (12) for $\mathcal{T}'$.

At the beginning of the algorithm we have to determine a global integral approximation based upon the initial $(d+1)$-pencil lattice on the simplicial partition $\mathcal{T}$ and then continue with the step (b). Since the step (c) is straightforward, we only have to describe steps (a) and (b). In the step (a) we select, for a given constant $\epsilon > 0$, collections of simplices $\{\triangle'_1, \triangle'_2, \ldots, \triangle'_{d+1}\}$, for which

$$\left| Q_{\triangle}(f) - \sum_{i=1}^{d+1} Q_{\triangle'_i}(f) \right| > \epsilon,$$

where $\triangle$ is the simplex that was split into $\triangle'_1, \triangle'_2, \ldots, \triangle'_{d+1}$ in the previous stage. Of course there are several ways how to perform step (b), which requires a subdivision of selected simplices (see [4], e.g.). But since our main goal is to keep the number of function evaluations at new points as low as possible, we will choose the subdivision strategy that will be based upon the lattice refinement approach, presented in Section 4. Recall that in this case we have to determine a subdivision point $\boldsymbol{T}$ inside the interior of a simplex, which defines $d+1$ new simplices

$$\triangle'_i = \langle \underbrace{\boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_{d-i}}_{d+1-i}, \underbrace{\boldsymbol{T}_{d-i+2}, \boldsymbol{T}_{d-i+3}, \ldots, \boldsymbol{T}_d}_{i-1}, \boldsymbol{T} \rangle, \quad i = 1, 2, \ldots, d+1. \tag{13}$$

Moreover, we have to determine the lattices on the newly obtained simplices, i.e., we have to choose a shape parameter ζ used by the lattice refinement (Fig. 5). Let us now consider two different possibilities how to determine a subdivision point and the lattices on new simplices.

Algorithm 1. We subdivide a simplex $\triangle = \langle \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \rangle$ to $d+1$ new simplices (13), where $\boldsymbol{T} := \frac{1}{d+1} \sum_{i=0}^{d} \boldsymbol{T}_i$, and the lattice on each simplex is a principal lattice. This is possible only if the lattices on all simplices in the original simplicial partition are principal lattices. We obtain the standard Newton-Cotes adaptive rule.

Algorithm 2. Using the lattice refinement approach presented in Section 4, we subdivide a simplex $\triangle = \langle \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \rangle$, having a $(d+1)$-pencil lattice determined by parameters $\boldsymbol{\xi} = (\xi_0, \xi_1, \ldots, \xi_d)$, to $d+1$ simplices $\{\triangle'_i\}_{i=1}^{d+1}$ given in (13), with lattices determined by parameters

$$
\begin{aligned}
&\boldsymbol{\xi}(\triangle'_1) = \left(\xi_0, \xi_1, \ldots, \xi_{d-2}, \zeta, \frac{\xi_{d-1}\xi_d}{\zeta}\right),\\
&\boldsymbol{\xi}(\triangle'_j) = \left(\xi_0, \xi_1, \ldots, \xi_{d-j-1}, \xi_{d-j} \cdot \xi_{d-j+1}, \xi_{d-j+2}, \ldots, \xi_{d-1}, \frac{\zeta}{\xi_{d-1}}, \frac{\xi_{d-1}\xi_d}{\zeta}\right),\\
&j = 2, 3, \ldots, d-1,\\
&\boldsymbol{\xi}(\triangle'_d) = \left(\xi_0 \cdot \xi_1, \xi_2, \xi_3, \ldots, \xi_{d-1}, \frac{\zeta}{\xi_{d-1}}, \frac{\xi_{d-1}\xi_d}{\zeta}\right),\\
&\boldsymbol{\xi}(\triangle'_{d+1}) = \left(\xi_1, \xi_2, \ldots, \xi_{d-1}, \frac{\zeta}{\xi_{d-1}}, \frac{\xi_{d-1}\xi_d\xi_0}{\zeta}\right),
\end{aligned}
\tag{14}
$$

where ζ is a free parameter. Suppose now that the number of function evaluations is crucial. The subdivision point $\boldsymbol{T} := \boldsymbol{z} = (z_1, z_2, \ldots, z_d)$ and the additional free parameter ζ in (14) can then be determined by a particular procedure, which can substantially decrease the number of function evaluations needed. Before we state this procedure, let us introduce the set of $\binom{n+d-1}{d-1}$ monomials of total degree n,

$$
\mathcal{P}_n := \left\{\boldsymbol{u}^{\boldsymbol{\gamma}},\ |\boldsymbol{\gamma}| = n,\ \boldsymbol{\gamma} \in \mathbb{N}_0^d\right\},
$$

and the closed ball $\Omega_\triangle \subset \triangle = \langle\, \boldsymbol{T}_0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_d \,\rangle \subset \mathbb{R}^d$,

$$
\Omega_\triangle := \Omega_\triangle(\boldsymbol{c}, r) := \{\boldsymbol{u} \in \triangle,\ \| \boldsymbol{u} - \boldsymbol{c} \|_2 \le r\}, \quad \boldsymbol{c} := \frac{1}{d+1}\sum_{i=0}^{d} \boldsymbol{T}_i,
$$

$$
r := \frac{1}{2} \cdot \min_{0 \le i < j \le d} \left\{ \left\| \frac{\boldsymbol{T}_i + \boldsymbol{T}_j}{2} - \boldsymbol{c} \right\|_2 \right\}.
$$

procedure ChooseParameters$(d, n, \triangle, \boldsymbol{\xi})$

1. value$(\boldsymbol{z}, \zeta) := 0$;
2. **for** $p \in \mathcal{P}_{n+1}$
3. sum$(\boldsymbol{z}, \zeta) = d! \sum_{i=1}^{d+1} \mathrm{vol}(\triangle'_i) \left(\sum_{|\boldsymbol{\gamma}|=n} \omega_{\boldsymbol{\gamma}}(\boldsymbol{\xi}(\triangle'_i))\, p(X_{\boldsymbol{\gamma}}(\triangle'_i))\right)$;
4. value$(\boldsymbol{z}, \zeta) =$ value$(\boldsymbol{z}, \zeta) + (S_\triangle(p) - \mathrm{sum}(\boldsymbol{z}, \zeta))^2$;
5. **end**;
6. $\{\bar{\boldsymbol{z}}, \bar{\zeta}\} := \left(\mathrm{value}(\bar{\boldsymbol{z}}, \bar{\zeta}) = \min_{\boldsymbol{z} \in \Omega_\triangle,\ \frac{1}{4} \le \zeta \le 4} \mathrm{value}(\boldsymbol{z}, \zeta)\right)$;
7. **Return** $\{\bar{\boldsymbol{z}}, \bar{\zeta}\}$;

Note that this procedure does not depend on the integrated function f. If our aim is to integrate several functions over the same simplicial partition, we will store all the computed parameters $\{\bar{\boldsymbol{z}}, \bar{\zeta}\}$.

Let us compare both algorithms on several interesting functions (Table 1, Fig. 6). Let $\mathcal{T} \subset [-1, 2] \times [-3/2, 3/2] \subset \mathbb{R}^2$ be a star (see [8], e.g.) and let $d = 2$ and $n = 3$. As expected, the number of function evaluations is significantly

smaller for the second algorithm, since there is a freedom of choosing the subdivision point $\boldsymbol{T}$ and the free parameter ζ at every step. Note that the choice of the parameter ζ brings approximately 20% to this fact.

Table 1
The number of function evaluations needed to achieve the error $|E(f)|$ is shown for both Newton-Cotes algorithms and for algorithms based on rules Q_G^4 and Q_G^6.

$d=2,\ n=3$			Alg. 1	Alg. 2	Q_G^4	Q_G^6
$f(u_1,u_2)$	$\int f$	$\lvert E(f)\rvert$	number of function evaluations			
$(u_1u_2)^3+(\cos(u_1+u_2))^2$	2.86003	0.005	**724**	**411**	**767**	**245**
$\cos(u_1+u_2)\,\sin(u_1+u_2)$	0.09144	0.0002	**1129**	**351**	**434**	**407**
$\cos(u_1+u_2)\,e^{\sin(u_1+u_2)}$	4.35648	0.0005	**832**	**471**	**308**	**254**
$e^{-\left((0.4\,(u_1-0.3))^2+0.2\,(u_2-0.2)^2\right)}$	5.51291	0.0001	**1345**	**651**	**488**	**317**
$e^{-(2\,\lvert u_1-0.2\rvert+0.6\,\lvert u_2+0.4\rvert)}+1$	8.33514	0.001	**1102**	**411**	**1217**	**1541**
$\frac{3}{2}u_1e^{-2\left((u_1-\frac{1}{2})^2+u_2^2\right)}-\frac{3}{2}u_2e^{-\left(u_1^2+u_2^2\right)}+1$	7.74886	0.0002	**3397**	**2211**	**1487**	**1046**

Let us conclude the paper with a brief efficiency comparison between particular Gaussian type adaptive formulae, and the cubature rules outlined in the paper. However, it should be emphasized that the computational complexity is not the only issue to be kept in mind when one is comparing these two classes of cubature rules. The Gaussian type requires the integrated function more or less to be known in a closed form. On the other hand, Newton-Cotes formulae, in the paper extended to generalized principal lattices, are closed form rules based upon the function values evaluated at particular unisolvent sets of points that can be simply generated in any dimension. Thus the data may be supplied in a tabular form only. Also, since it is straightforward to generate the lattice points, a way to a working computer program may be shorter. As for the numerical test, let us recompute the examples by a similar adaptive algorithm, but based upon two different Gaussian rules for $d=2$ and $n=3$. In the barycentric form, the weights and the points of these two rules are (see [3], e.g.)

$$Q_G^4:\qquad \omega_i=\frac{25}{96},\ i=1,2,3,\quad \omega_4=-\frac{9}{32},$$
$$B_1=\left(\frac{3}{5},\frac{1}{5},\frac{1}{5}\right),\ B_2=\left(\frac{1}{5},\frac{3}{5},\frac{1}{5}\right),\ B_3=\left(\frac{1}{5},\frac{1}{5},\frac{3}{5}\right),\ B_4=\left(\frac{1}{3},\frac{1}{3},\frac{1}{3}\right),$$

$$Q_G^6:\qquad \omega_i=\frac{1}{12},\ i=1,2,\ldots,6,$$
$$\tau_1:=0.10903900907288,\ \tau_2:=0.23193336855303,\ \tau_3:=1-\tau_1-\tau_2,$$
$$B_1=(\tau_1,\tau_2,\tau_3),\ B_2=(\tau_1,\tau_3,\tau_2),\ B_3=(\tau_2,\tau_1,\tau_3),$$
$$B_4=(\tau_2,\tau_3,\tau_1),\ B_5=(\tau_3,\tau_1,\tau_2),\ B_6=(\tau_3,\tau_2,\tau_1).$$

The number of function evaluations in the adaptive Gaussian algorithms based upon these two rules are shown in Table 1.

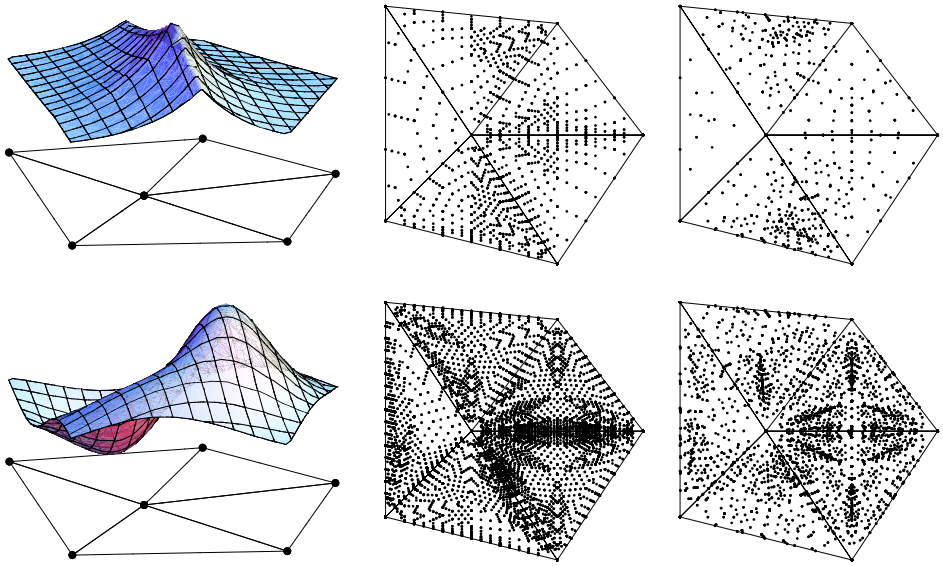

Fig. 6. The points where the function evaluations are needed in both Newton-Cotes algorithms for the last two rows in Table 1.

References

[1] J.M. Carnicer, M. Gasca, T. Sauer, Interpolation lattices in several variables, *Numer. Math.* **102** (2006) 559–581.

[2] K.C. Chung, T.H. Yao, On lattices admitting unique Lagrange interpolation, *SIAM J. Numer. Anal.* **14** (1977) 735–743.

[3] R. Cools, An Encyclopaedia of Cubature Formulas, *J. Complexity* **19** (2003) 445–453.

[4] R. Cools, A. Genz, An adaptive numerical cubature algorithm for simplices, *ACM Trans. Math. Software* **29** (2003) 297–308.

[5] G. Jaklič, J. Kozak, M. Krajnc, V. Vitrih, E. Žagar, Three-pencil lattices on triangulations, *Numer. Algor.* 45 (2007) 49–60.

[6] G. Jaklič, J. Kozak, M. Krajnc, V. Vitrih, E. Žagar, Barycentric coordinates for Lagrange interpolation over lattices on a simplex, *Numer. Algor.* **4**8 (2008) 93–104.

[7] G. Jaklič, J. Kozak, M. Krajnc, V. Vitrih, E. Žagar, Lattices on simplicial partitions, to appear in J. Comput. Appl. Math.

[8] M.J. Lai, L.L. Schumaker, *Spline functions on triangulations*, Encyclopedia of mathematics and its applications, Cambridge University Press, 2007.

[9] S.L. Lee, G.M. Phillips, Construction of lattices for Lagrange interpolation in projective space, *Constr. Approx.* **7** (1991) 283–297.

[10] R.A. Nicolaides, On a class of finite elements generated by Lagrange interpolation, *SIAM J. Numer. Anal.* **9** (1972) 435–445.

[11] T. Sauer, Y. Xu, On multivariate Lagrange interpolation, *Math. Comp.* **64** (1995) 1147–1170.

[12] A.H. Stroud, *Approximate calculation of multiple integrals*, Prentice-Hall, Englewood Cliffs, N.J., 1971.

[13] V. Vitrih, Lattices on tetrahedral partitions, *Annali dell'Universita di Ferrara* **54** (2008) 349–359.